Application No. 10/792,154
AF Response dated November 21, 2006
Final Office Action dated June 21, 2006

AMENDMENTS TO THE CLAIMS:

This listing of claims will replace all prior versions, and listings, of claims in the application.

1-16. (Canceled)

17. (Previously Presented) A method comprising:

initially allocating execution resources for multiple threads;

determining that a first thread has stalled;

temporarily storing one or more instructions of the first thread in a replay queue; and

continuing to allocate execution resources to other threads which have not stalled.

18. (Original) The method of claim 17 wherein said step of continuing to allocate comprises the step of continuing to allocate execution resources to the other threads which have not stalled and inhibiting the allocation of further resources to the stalled thread by temporarily storing the stalled thread instructions in the replay queue.

19. (Original) The method of claim 17 wherein priority for execution resources are allocated to the other threads which have not stalled on a rotating priority basis.

20. (Original) The method of claim 17 and further comprising the steps of:

detecting that the first thread is no longer stalled;

unloading the one or more instructions of the first thread from the replay queue; and

re-allocating at least some execution resources to the first thread.

21. (Original) The method of claim 17 wherein the step of determining that a first thread has stalled comprises detecting a long latency or agent instruction for the first thread.

22. (Original) The method of claim 17, further comprising routing the one or more instructions of the first thread stored in the replay queue to an execution unit for re-executing the one or more instructions of the first thread.

23. (Original) The method of claim 17, further comprising:

determining that a second thread has stalled;

temporarily storing one or more instructions of the second thread; and

continuing to allocate execution resources to other threads which have not stalled.

24-34. (Canceled)

35. (Original) A method comprising:

allocating execution resources for multiple threads;

determining that a first thread has stalled;

storing one or more instructions of the first thread in a first thread replay queue section;

determining that a second thread has stalled; and

storing one or more instructions of the second thread in a second thread replay queue section.

36. (Canceled)

37. (Previously Presented) The method of claim 35 wherein priority for execution resources are allocated to the other threads which have not stalled on a rotating priority basis.

38. (Previously Presented) The method of claim 35 and further comprising the steps of:

detecting that the first thread is no longer stalled;

unloading the one or more instructions of the first thread from the replay queue; and

re-allocating at least some execution resources to the first thread.

39. (Previously Presented) The method of claim 35 wherein the step of determining that a first thread has stalled comprises detecting a long latency or agent instruction for the first thread.

40. (Previously Presented) The method of claim 35 further comprising routing the one or more instructions of the first thread stored in the replay queue to an execution unit for re-executing the one or more instructions of the first thread.